Tucker Tensor Regression
and Neuroimaging Analysis

Xiaoshan Li, Hua Zhou and Lexin Li

North Carolina State University

Abstract

Large-scale neuroimaging studies have been collecting brain images of study individuals, which take the form of two-dimensional, three-dimensional, or higher dimensional arrays, also known as tensors. Addressing scientific questions arising from such data demands new regression models that take multidimensional arrays as covariates. Simply turning an image array into a long vector causes extremely high dimensionality that compromises classical regression methods, and, more seriously, destroys the inherent spatial structure of array data that possesses wealth of information. In this article, we propose a family of generalized linear tensor regression models based upon the Tucker decomposition of regression coefficient arrays. Effectively exploiting the low rank structure of tensor covariates brings the ultrahigh dimensionality to a manageable level that leads to efficient estimation. We demonstrate, both numerically that the new model could provide a sound recovery of even high rank signals, and asymptotically that the model is consistently estimating the best Tucker structure approximation to the full array model in the sense of Kullback-Liebler distance. The new model is also compared to a recently proposed tensor regression model that relies upon an alternative CANDECOMP/PARAFAC (CP) decomposition.

Key Words: CP decomposition; magnetic resonance image; tensor; Tucker decomposition.

1   Introduction

Advancing technologies are constantly producing large scale scientific data with complex structures. An important class arises from medical imaging, where the data takes the form of multidimensional array, also known as *tensor*. Notable examples include electroencephalography (EEG, 2D matrix), anatomical magnetic resonance images (MRI, 3D array), functional magnetic resonance images (fMRI, 4D array), among other image modalities. In medical

---

[1]*Address for correspondence: Lexin Li, Department of Statistics, North Carolina State University, Box 8203, Raleigh, NC 27695-8203. Email: lexin_li@ncsu.edu.*

1

imaging data analysis, a primary goal is to better understand associations between brains and clinical outcomes. Applications include using brain images to diagnose neurodegenerative disorders, to predict onset of neuropsychiatric diseases, and to identify disease relevant brain regions or activity patterns. This family of problems can collectively be formulated as a regression with clinical outcome as response, and image, or tensor, as predictor. However, the sheer size and complex structure of image covariate pose unusual challenges, which motivate us to develop a new class of regression models with image covariate.

Most classical regression models take vector as covariate. Naively turning an image array into a vector is evidently unsatisfactory. For instance, a typical MRI image of size 128-by-128-by-128 implicitly requires $128^3 = 2,097,152$ regression parameters. Both computability and theoretical guarantee of the classical regression models are severely compromised by this ultra-high dimensionality. More seriously, vectorizing an array destroys the inherent spatial structure of the image array that usually possesses abundant information. A typical solution in the literature first employs the subject knowledge to extract a vector of features from images, and then feeds the feature vector into a classical regression model (Mckeown et al., 1998; Blankertz et al., 2001; Haxby et al., 2001; Kontos et al., 2003; Mitchell et al., 2004; LaConte et al., 2005; Shinkareva et al., 2006). Alternatively one first applies unsupervised dimension reduction, often some variant of principal components analysis, to the image array, and then fits a regression model in the reduced dimensional vector space (Caffo et al., 2010). Both solutions are intuitive and popular, and have enjoyed varying degrees of success. At heart, both transform the problem to a classical vector covariate regression. However, there is no consensus on what choice best summarizes a brain image even for a single modality, whereas unsupervised dimension reduction like principal components could result in information loss in a regression setup. In contrast to constructing an image feature vector, the functional approach views image as a function and then employs functional regression models (Ramsay and Silverman, 2005). Reiss and Ogden (2010) notably applied this idea to regression with 2D image predictor. Extending their method to 3D and higher dimensional images, however, is far from trivial and requires substantial research, due to the large number of parameters and multi-collinearity among imaging measures.

In a recent work, Zhou et al. (2013) proposed a class of generalized linear *tensor* regression models. Specifically, for a response variable Y, a vector predictor $\boldsymbol{Z} \in \mathbb{R}^{p_0}$ and a D-

dimensional tensor predictor $\boldsymbol{X} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$, the response is assumed to belong to an exponential family where the linear systematic part is of the form,

$$g(\mu) = \boldsymbol{\gamma}^{\mathsf{T}} \boldsymbol{Z} + \langle \boldsymbol{B}, \boldsymbol{X} \rangle. \tag{1}$$

Here $g(\cdot)$ is a strictly increasing link function, $\mu = E(Y|\boldsymbol{X}, \boldsymbol{Z})$, $\boldsymbol{\gamma} \in \mathbb{R}^{p_0}$ is the regular regression coefficient vector, $\boldsymbol{B} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$ is the coefficient array that captures the effects of tensor covariate $\boldsymbol{X}$, and the inner product between two arrays is defined as $\langle \boldsymbol{B}, \boldsymbol{X} \rangle = \langle \text{vec}\boldsymbol{B}, \text{vec}\boldsymbol{X} \rangle = \sum_{i_1,\ldots,i_D} \beta_{i_1\ldots i_D} x_{i_1\ldots i_D}$. This model, if with no further simplification, is prohibitive given its gigantic dimensionality: $p_0 + \prod_{d=1}^{D} p_d$. Motivated by a commonly used tensor decomposition, Zhou et al. (2013) introduced a low rank structure on the coefficient array $\boldsymbol{B}$. That is, $\boldsymbol{B}$ is assumed to follow a rank-R CANDECOMP/PARAFAC (CP) decomposition (Kolda and Bader, 2009),

$$\boldsymbol{B} = \sum_{r=1}^{R} \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r)}, \tag{2}$$

where $\boldsymbol{\beta}_d^{(r)} \in \mathbb{R}^{p_d}$ are all column vectors, $d = 1, \ldots, D, r = 1, \ldots, R$, and $\circ$ denotes an outer product among vectors. Here the outer product $\boldsymbol{b}_1 \circ \boldsymbol{b}_2 \circ \cdots \circ \boldsymbol{b}_D$ of D vectors $\boldsymbol{b}_d \in \mathbb{R}^{p_d}$, $d = 1, \ldots, D$, is defined as the $p_1 \times \cdots \times p_D$ array with entries $(\boldsymbol{b}_1 \circ \boldsymbol{b}_2 \circ \cdots \circ \boldsymbol{b}_D)_{i_1 \cdots i_D} = \prod_{d=1}^{D} b_{d i_d}$. For convenience, this CP decomposition is often represented by a shorthand $\boldsymbol{B} = [\![\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D]\!]$, where $\boldsymbol{B}_d = [\boldsymbol{\beta}_d^{(1)}, \ldots, \boldsymbol{\beta}_d^{(R)}] \in \mathbb{R}^{p_d \times R}$, $d = 1, \ldots, D$. Combining (1) and (2) yields generalized linear tensor regression models of Zhou et al. (2013), where the dimensionality decreases to the scale of $p_0 + R \times \sum_{d=1}^{D} p_d$. Under this setup, ultrahigh dimensionality of (1) is reduced to a manageable level, which in turn results in efficient estimation and prediction. For instance, for a regression with 128-by-128-by-128 MRI image and 5 usual covariates, the dimensionality is reduced from the order of $2,097,157 = 5 + 128^3$ to $389 = 5 + 128 \times 3$ for a rank-1 model, and to $1,157 = 5 + 3 \times 128 \times 3$ for a rank-3 model. Zhou et al. (2013) showed that this low rank tensor model could provide a sound recovery of even high rank signals.

In the tensor literature, there has been an important development parallel to CP decomposition, which is called Tucker decomposition, or higher-order singular value decomposition (HOSVD) (Kolda and Bader, 2009). In this article, we propose a class of *Tucker tensor regression models*. To differentiate, we call the models of Zhou et al. (2013) *CP tensor*

3

regression models. Specifically, we continue to adopt the model (1), but assume that the coefficient array $\boldsymbol{B}$ follows a Tucker decomposition,

$$\boldsymbol{B} = \sum_{r_1=1}^{R_1} \cdots \sum_{r_D=1}^{R_D} g_{r_1,\ldots,r_D} \boldsymbol{\beta}_1^{(r_1)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r_D)}, \tag{3}$$

where $\boldsymbol{\beta}_d^{(r_d)} \in \mathbb{R}^{p_d}$ are all column vectors, $d = 1, \ldots, D, r_d = 1, \ldots, R_d$, and $g_{r_1,\ldots,r_D}$ are constants. It is often abbreviated as $\boldsymbol{B} = [\![\boldsymbol{G}; \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D]\!]$, where $\boldsymbol{G} \in \mathbb{R}^{R_1 \times \cdots \times R_D}$ is a D-dimensional *core tensor* with entries $(\boldsymbol{G})_{r_1 \ldots r_D} = g_{r_1,\ldots,r_D}$, and $\boldsymbol{B}_d \in \mathbb{R}^{p_d \times R_d}$ are the factor matrices. $\boldsymbol{B}_d$'s are usually orthogonal and can be thought of as the *principal components* in each dimension (and thus the name, HOSVD). The number of parameters of a Tucker tensor model is in the order of $p_0 + \sum_{d=1}^{D} R_d \times p_d$. Comparing the two decompositions (2) and (3), the key difference is that CP fixes the number of basis vectors R along each dimension of $\boldsymbol{B}$ so that all $\boldsymbol{B}_d$'s have the *same* number of columns (ranks). In contrast, Tucker allows the number R_d to differ along different dimensions and $\boldsymbol{B}_d$'s could have *different* ranks.

This difference between the two decompositions seems minor; however, in the context of tensor regression modeling and neuroimging analysis, it has profound implications, and such implications motivate this article. On one hand, the Tucker tensor regression model shares the advantages of the CP tensor regression model, in that it effectively exploits the special structure of the tensor data, it substantially reduces the dimensionality to enable efficient model estimation, and it provides a sound low rank approximation to a potentially high rank signal. On the other hand, Tucker tensor regression offers a much more *flexible* modeling framework than CP regression, as it allows distinct order along each dimension. When the orders are all identical, it includes the CP model as a special case. This flexibility leads to several improvements that are particularly useful for neuroimaging analysis. First, a Tucker model could be more parsimonious than a CP model thanks to the flexibility of different orders. For instance, suppose a 3D signal $\boldsymbol{B} \in \mathbb{R}^{16 \times 16 \times 16}$ admits a Tucker decomposition (3) with $R_1 = R_2 = 2$ and $R_3 = 5$. It can only be recovered by a CP decomposition with $R = 5$, costing 230 parameters. In contrast, the Tucker model is more parsimonious with only 131 parameters. This reduction of free parameters is valuable for medical imaging studies, as the number of subjects is often limited. Second, the freedom in the choice of different orders is useful when the tensor data is skewed in dimensions, which is common in neuroimaging data. For instance, in EEG, the two dimensions consist of electrodes (channels) and time,

and the number of sampling time points usually far exceeds the number of channels. Third, even when all tensor modes have comparable sizes, the Tucker formulation explicitly models the interactions between factor matrices $\boldsymbol{B}_d$'s, and as such allows a finer grid search within a larger model space, which in turn may explain more trait variance. Finally, as we will show in Section 2.3, there exists a duality regarding the Tucker tensor model. Thanks to this duality, a Tucker tensor decomposition naturally lends itself to a principled way of imaging data downsizing, which, given the often limited sample size, again plays a practically very useful role in neuroimaging analysis.

For these reasons, we feel it important to develop a complete methodology of Tucker tensor regression and its associated theory. The resulting Tucker tensor model carries a number of useful features. It performs dimension reduction through low rank tensor decomposition but in a supervised fashion, and as such avoids potential information loss in regression. It works for general array-valued image modalities and/or any combination of them, and for various types of responses, including continuous, binary, and count data. Besides, an efficient and highly scalable algorithm has been developed for the associated maximum likelihood estimation. This scalability is important considering the massive scale of imaging data. In addition, regularization has been studied in conjunction with the proposed model, yielding a collection of regularized Tucker tensor models, and particularly one that encourages sparsity of the core tensor to facilitate model selection among the defined Tucker model space.

Recently there have been some increasing interests in matrix/tensor decomposition and their applications in brain imaging studies (Crainiceanu et al., 2011; Allen et al., 2011; Hoff, 2011; Aston and Kirch, 2012). Nevertheless, this article is distinct in that we concentrate on a regression framework with scalar response and tensor valued covariates. In contrast, Crainiceanu et al. (2011) and Allen et al. (2011) studied unsupervised decomposition, Hoff (2011) considered model-based decomposition, whereas Aston and Kirch (2012) focused on change point distribution estimation. The most closely related work to this article is Zhou et al. (2013); however, we feel our work is *not* a simple extension of theirs. First of all, considering the complex nature of tensor, the development of the Tucker model estimation as well as its asymptotics is far from a trivial extension of the CP model of Zhou et al. (2013). Moreover, we offer a detailed comparison, both analytically (in Section 2.4) and numerically (in Sections 6.3 and 6.4), of the CP and Tucker decompositions in the context

of regression with imaging/tensor covariates. We believe this comparison is crucial for an adequate comprehension of tensor regression models and supervised tensor decomposition in general.

The rest of the article is organized as follows. Section 2 begins with a brief review of some preliminaries on tensor, and then presents the Tucker tensor regression model. Section 3 develops an efficient algorithm for maximum likelihood estimation. Section 4 derives inferential tools such as score, Fisher information, identifiability, consistency, and asymptotic normality. Section 5 investigates regularization method for the Tucker regression. Section 6 presents extensive numerical results. Section 7 concludes with some discussions and points to future extensions. All technical proofs are delegated to the Appendix.

2 Model

2.1 Preliminaries

We start with a brief review of some matrix/array operations and results. Extensive references can be found in the survey paper (Kolda and Bader, 2009).

A *tensor* is a multidimensional array. *Fibers* of a tensor are the higher order analogue of matrix rows and columns. A fiber is defined by fixing every index but one. A matrix column is a mode-1 fiber and a matrix row is a mode-2 fiber. Third-order tensors have column, row, and tube fibers, respectively. We next review some important operators that transform a tensor into a vector/matrix. The *vec operator* stacks the entries of a D-dimensional tensor $\boldsymbol{B} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$ into a column vector. Specifically, an entry $b_{i_1 \ldots i_D}$ maps to the j-th entry of $\text{vec}\,\boldsymbol{B}$ where $j = 1 + \sum_{d=1}^{D}(i_d - 1)\prod_{d'=1}^{d-1} p_{d'}$. For instance, when $D = 2$, the matrix entry at cell (i_1, i_2) maps to position $j = 1 + i_1 - 1 + (i_2 - 1)p_1 = i_1 + (i_2 - 1)p_1$, which is consistent with the more familiar vec operator on a matrix. The *mode-d matricization*, $\boldsymbol{B}_{(d)}$, maps a tensor $\boldsymbol{B}$ into a $p_d \times \prod_{d' \neq d} p_{d'}$ matrix such that the $(i_1, \ldots, i_D)$ element of the array $\boldsymbol{B}$ maps to the (i_d, j) element of the matrix $\boldsymbol{B}_{(d)}$, where $j = 1 + \sum_{d' \neq d}(i_{d'} - 1)\prod_{d'' < d', d'' \neq d} p_{d''}$. When $D = 1$, we observe that $\text{vec}\,\boldsymbol{B}$ is the same as vectorizing the mode-1 matricization $\boldsymbol{B}_{(1)}$. The *mode-(d, d') matricization* $\boldsymbol{B}_{(dd')} \in \mathbb{R}^{p_d p_{d'} \times \prod_{d'' \neq d, d'} p_{d''}}$ is defined in a similar fashion. We then define the *mode-d multiplication* of the tensor $\boldsymbol{B}$ with a matrix $\boldsymbol{U} \in \mathbb{R}^{p_d \times q}$, denoted by $\boldsymbol{B} \times_d \boldsymbol{U} \in \mathbb{R}^{p_1 \times \cdots \times q \times \cdots \times p_D}$, as the multiplication of the mode-d fibers of $\boldsymbol{B}$ by $\boldsymbol{U}$. In other

words, the mode-d matricization of $\boldsymbol{B} \times_d \boldsymbol{U}$ is $\boldsymbol{U}\boldsymbol{B}_{(d)}$.

We also review two properties of a tensor $\boldsymbol{B}$ that admits a Tucker decomposition (3). The mode-d matricization of $\boldsymbol{B}$ can be expresses as

$$\boldsymbol{B}_{(d)} = \boldsymbol{B}_d \boldsymbol{G}_{(d)} (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1)^\mathsf{T},$$

where $\otimes$ denotes the Kronecker product of matrices. If applying the vec operator to $\boldsymbol{B}$, then

$$\mathrm{vec}\boldsymbol{B} = \mathrm{vec}\boldsymbol{B}_{(1)} = \mathrm{vec}(\boldsymbol{B}_1 \boldsymbol{G}_{(1)} (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_2)^\mathsf{T}) = (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1)\mathrm{vec}\boldsymbol{G}.$$

These two properties are useful for our subsequent Tucker regression development.

2.2 Tucker Regression Model

We elaborate on the Tucker tensor regression model introduced in Section 1. We assume that Y belongs to an exponential family with probability mass function or density (McCullagh and Nelder, 1983),

$$p(y_i|\theta_i, \phi) = \exp\left\{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right\}$$

with the first two moments $E(Y_i) = \mu_i = b'(\theta_i)$ and $\mathrm{Var}(Y_i) = \sigma_i^2 = b''(\theta_i)a_i(\phi)$. θ and $\phi > 0$ are, respectively, called the natural and dispersion parameters. We assume the systematic part of GLM is of the form

$$g(\mu) = \eta = \boldsymbol{\gamma}^\mathsf{T}\boldsymbol{Z} + \langle \sum_{r_1=1}^{R_1} \cdots \sum_{r_D=1}^{R_D} g_{r_1,\ldots,r_D}\boldsymbol{\beta}_1^{(r_1)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r_D)}, \boldsymbol{X}\rangle. \tag{4}$$

That is, we impose a Tucker structure on the array coefficient $\boldsymbol{B}$. We make a few remarks. First, in this article, we consider the problem of estimating the core tensor $\boldsymbol{G}$ and factor matrices $\boldsymbol{B}_d$ simultaneously given the response Y and covariates $\boldsymbol{X}$ and $\boldsymbol{Z}$. This can be viewed as a *supervised* version of the classical unsupervised Tucker decomposition. It is also a supervised version of principal components analysis for higher-order multidimensional array. Unlike a two-stage solution that first performs principal components analysis and then fits a regression model, the basis (principal components) $\boldsymbol{B}_d$ in our models are estimated under the guidance (supervision) of the response variable. Second, the CP model of Zhou et al. (2013) corresponds to a special case of the Tucker model (4) with $g_{r_1,\ldots,r_D} = 1_{\{r_1=\cdots=r_D\}}$ and $R_1 = \ldots = R_D = R$. In other words, the CP model is a specific Tucker model with

a super-diagonal core tensor $\boldsymbol{G}$. The CP model has a rank at most R while the general Tucker model can have a rank as high as R^D. We will further compare the two model sizes in Section 2.4.

2.3 Duality and Tensor Basis Pursuit

Next we investigate a duality regarding the inner product between a general tensor and a tensor that admits a Tucker decomposition.

Lemma 1 (Duality). *Suppose a tensor $\boldsymbol{B} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$ admits Tucker decomposition $\boldsymbol{B} = [\![\boldsymbol{G}; \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D]\!]$. Then, for any tensor $\boldsymbol{X} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$, $\langle \boldsymbol{B}, \boldsymbol{X} \rangle = \langle \boldsymbol{G}, \tilde{\boldsymbol{X}} \rangle$, where $\tilde{\boldsymbol{X}}$ admits a Tucker decomposition $\tilde{\boldsymbol{X}} = [\![\boldsymbol{X}; \boldsymbol{B}_1^\intercal, \ldots, \boldsymbol{B}_D^\intercal]\!]$.*

This duality gives some important insights to the Tucker tensor regression model. First, if we consider $\boldsymbol{B}_d \in \mathbb{R}^{p_d \times R_d}$ as fixed and known basis matrices, then Lemma 1 says fitting the Tucker tensor regression model (4) is equivalent to fitting a tensor regression model in $\boldsymbol{G}$ with the *transformed* data $\tilde{\boldsymbol{X}} = [\![\boldsymbol{X}; \boldsymbol{B}_1^\intercal, \ldots, \boldsymbol{B}_D^\intercal]\!] \in \mathbb{R}^{R_1 \times \cdots \times R_D}$. When $R_d \ll p_d$, the transformed data $\tilde{\boldsymbol{X}}$ effectively *downsize* the original data. We will further illustrate this downsizing feature in the real data analysis in Section 6.4. Second, in applications where the numbers of basis vectors R_d are unknown, we can utilize possibly over-complete basis matrices $\boldsymbol{B}_d$ such that $R_d \geq p_d$, and then estimate $\boldsymbol{G}$ with sparsity regularizations. This leads to a tensor version of the classical basis pursuit problem (Chen et al., 2001). Take fMRI data as an example. We can adopt the wavelet basis for the three image dimensions and the Fourier basis for the time dimension. Regularization on $\boldsymbol{G}$ can be achieved by either imposing a low rank decomposition (CP or Tucker) on $\boldsymbol{G}$ (hard thresholding) or penalized regression (soft thresholding). We will investigate Tucker regression regularization in details in Section 5.

2.4 Model Size: Tucker vs CP

In this section we investigate the size of the Tucker tensor model. Comparison with the size of the CP tensor model helps gain better understanding of both models. In addition, it provides a base for data adaptive selection of appropriate orders in a Tucker model.

First we quickly review the number of free parameters p_{C} for a CP model $\boldsymbol{B} = [\![\boldsymbol{B}_1, \ldots, \boldsymbol{B}_d]\!]$, with $\boldsymbol{B}_d \in \mathbb{R}^{p_d \times R}$. For $D = 2$, $p_{\mathrm{C}} = R(p_1 + p_2) - R^2$, and for $D > 2$, $p_{\mathrm{C}} = R(\sum_{d=1}^{D} p_d - D + 1)$.

Table 1: Number of free parameters in Tucker and CP models.

	CP	Tucker
$D = 2$	$R(p_1 + p_2) - R^2$	$p_1 R_1 + p_2 R_2 + R_1 R_2 - R_1^2 - R_2^2$
$D > 2$	$R(\sum_d p_d - D + 1)$	$\sum_d p_d R_d + \prod_d R_d - \sum_d R_d^2$

For $D = 2$, the term $-R^2$ adjusts for the nonsingular transformation indeterminacy for model identifiability; for $D > 2$, the term $R(-D + 1)$ adjusts for the scaling indeterminacy in the CP decomposition. See Zhou et al. (2013) for more details. Following similar arguments, we obtain that the number of free parameters p_T in a Tucker model $\boldsymbol{B} = [\![\boldsymbol{G}; \boldsymbol{B}_1, \ldots, \boldsymbol{B}_d]\!]$, with $\boldsymbol{G} \in \mathbb{R}^{R_1 \times \cdots \times R_d}$ and $\boldsymbol{B}_d \in \mathbb{R}^{p_d \times R_d}$, is

$$p_T = \sum_{d=1}^{D} p_d R_d + \prod_{d=1}^{D} R_d - \sum_{d=1}^{D} R_d^2,$$

for any D. Here the term $-\sum_{d=1}^{D} R_d^2$ adjusts for the non-singular transformation indeterminancy in the Tucker decomposition. We summarize these results in Table 1.

Next we compare the two model sizes (degrees of freedom) under an additional assumption that $R_1 = \cdots = R_d = R$. The difference becomes:

$$p_T - p_C = \begin{cases} 0 & \text{when } D = 2, \\ R(R-1)(R-2) & \text{when } D = 3, \\ R(R^3 - 4R + 3) & \text{when } D = 4, \\ R(R^{D-1} - DR + D - 1) & \text{when } D > 4. \end{cases}$$

Based on this formula, when $D = 2$, the Tucker model is essentially the same as the CP model. When $D = 3$, Tucker has the same number of parameters as CP for $R = 1$ or $R = 2$, but costs $R(R-1)(R-2)$ more parameters for $R > 2$. When $D > 3$, Tucker and CP are the same for $R = 1$, but Tucker costs substantially more parameters than CP for $R > 2$. For instance, when $D = 4$ and $R = 3$, Tucker model takes 54 more parameters than the CP model. However, one should bear in mind that the above discussion assumes $R_1 = \cdots = R_d = R$. In reality, Tucker could require *less* free parameters than CP, as shown in the illustrative example given in Section 1, since Tucker is more flexible and allows different order R_d along each dimension.

Figure 1 shows an example with $D = 3$ dimensional array covariates. Half of the true signal (brain activity map) $\boldsymbol{B}$ is displayed in the left panel, which is by no means a low rank
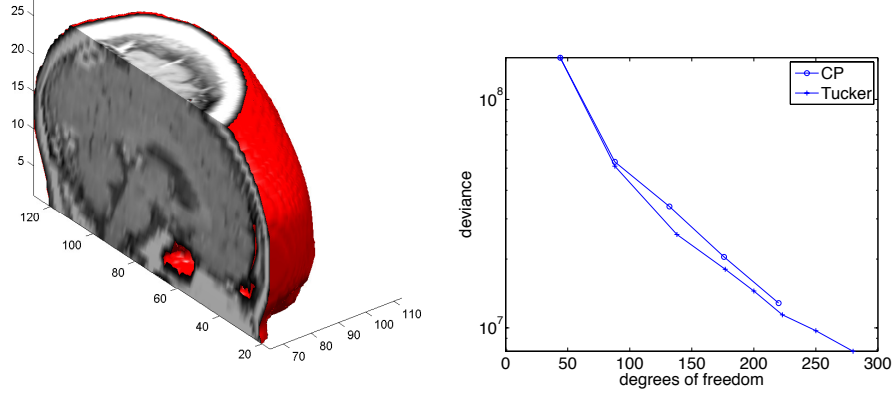
Figure 1: Left: half of the true signal array $\boldsymbol{B}$. Right: Deviances of CP regression estimates at $R = 1, \ldots, 5$, and Tucker regression estimates at orders $(R_1, R_2, R_3) = (1,1,1), (2,2,2),$ $(3,3,3), (4,4,3), (4,4,4), (5,4,4), (5,5,4),$ and $(5,5,5)$. The sample size is $n = 1000$.

signal. Suppose 3D images $\boldsymbol{X}_i$ are taken on $n = 1,000$ subjects. We simulate image traits $\boldsymbol{X}_i$ from independent standard normals and quantitative traits Y_i from independent normals with mean $\langle \boldsymbol{X}_i, \boldsymbol{B} \rangle$ and unit variance. Given the limited sample size, the hope is to infer a reasonable low rank approximation to the activity map from the 3D image covariates. The right panel displays the model deviance versus the degrees of freedom of a series of CP and Tucker model estimates. The CP model is estimated at ranks $R = 1, \ldots, 5$. The Tucker model is fitted at orders $(R_1, R_2, R_3) = (1,1,1), (2,2,2), (3,3,3), (4,4,3), (4,4,4), (5,4,4),$ $(5,5,4),$ and $(5,5,5)$. We see from the plot that, under the same number of free parameters, the Tucker model could generally achieve a better model fit with a smaller deviance. (Note that the deviance is in the log scale, so a small discrepancy between the two lines translates to a large value of difference in deviance.)

The explicit model size formula of the Tucker model is also useful for choosing appropriate orders R_d's along each direction given data. This can be treated as a model selection problem, and we can employ a typical model selection criterion, e.g., Bayesian information criterion (BIC). It is of the form: $-2 \log \ell + \log(n) p_e$, where ℓ is the log-likelihood, and $p_e = p_{\mathrm{T}}$ is the effective number of parameters of the Tucker model as given in Table 1. We will illustrate this BIC criterion in the numerical Section 6.1, and will discuss some heuristic guidelines of selecting orders in Section 6.4.

10

3 Estimation

We pursue the maximum likelihood estimation (MLE) for the Tucker tensor regression model and develop a scalable estimation algorithm in this section. The key observation is that, although the systematic part (4) is not linear in $\boldsymbol{G}$ and $\boldsymbol{B}_d$ *jointly*, it is linear in them *separately*. This naturally suggests a block relaxation algorithm, which updates each factor matrix $\boldsymbol{B}_d$ and the core tensor $\boldsymbol{G}$ *alternately*.

The algorithm consists of two core steps. First, when updating $\boldsymbol{B}_d \in \mathbb{R}^{p_d \times R_d}$ with the rest $\boldsymbol{B}_{d'}$'s and $\boldsymbol{G}$ fixed , we rewrite the array inner product in (4) as

$$
\begin{aligned}
\langle \boldsymbol{B}, \boldsymbol{X} \rangle &= \langle \boldsymbol{B}_{(d)}, \boldsymbol{X}_{(d)} \rangle \\
&= \langle \boldsymbol{B}_d \boldsymbol{G}_{(d)} (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1)^\mathsf{T}, \boldsymbol{X}_{(d)} \rangle \\
&= \langle \boldsymbol{B}_d, \boldsymbol{X}_{(d)} (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1) \boldsymbol{G}_{(d)}^\mathsf{T} \rangle.
\end{aligned}
$$

Then the problem turns into a GLM regression with $\boldsymbol{B}_d$ as the "parameter" and the term $\boldsymbol{X}_{(d)} (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1) \boldsymbol{G}_{(d)}^\mathsf{T}$ as the "predictor". It is a low dimensional GLM with only $p_d R_d$ parameters and thus is easy to solve. Second, when updating $\boldsymbol{G} \in \mathbb{R}^{R_1 \times \cdots \times R_D}$ with all $\boldsymbol{B}_d$'s fixed,

$$
\begin{aligned}
\langle \boldsymbol{B}, \boldsymbol{X} \rangle &= \langle \mathrm{vec} \boldsymbol{B}, \mathrm{vec} \boldsymbol{X} \rangle \\
&= \langle (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1) \mathrm{vec} \boldsymbol{G}, \mathrm{vec} \boldsymbol{X} \rangle \\
&= \langle \mathrm{vec} \boldsymbol{G}, (\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1)^\mathsf{T} \mathrm{vec} \boldsymbol{X} \rangle.
\end{aligned}
$$

This implies a GLM regression with $\mathrm{vec} \boldsymbol{G}$ as the "parameter" and the term $(\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1)^\mathsf{T} \mathrm{vec} \boldsymbol{X}$ as the "predictor". Again this is a low dimensional regression problem with $\prod_d R_d$ parameters. For completeness, we summarize the above alternating estimation procedure in Algorithm 1. The orthogonality between the columns of factor matrices $\boldsymbol{B}_d$ is not enforced as in unsupervised HOSVD, because our primary goal is approximating tensor signal instead of finding the principal components along each mode.

Next we study the convergence properties of the proposed algorithm. As the block relaxation algorithm monotonically increases the objective value, the stopping criterion is well-defined and the convergence properties of iterates follow from the standard theory for monotone algorithms (de Leeuw, 1994; Lange, 2010). The proof of next result is given in the Appendix.

Algorithm 1 Block relaxation algorithm for fitting the Tucker tensor regression.

Initialize: $\boldsymbol{\gamma}^{(0)} = \mathrm{argmax}_{\boldsymbol{\gamma}} \, \ell(\boldsymbol{\gamma}, \mathbf{0}, \ldots, \mathbf{0})$, $\boldsymbol{B}_d^{(0)} \in \mathbb{R}^{p_d \times R_d}$ a random matrix for $d = 1, \ldots, D$,
and $\boldsymbol{G}^{(0)} \in \mathbb{R}^{R_1 \times \cdots \times R_D}$ a random matrix.
repeat
    for $d = 1, \ldots, D$ **do**
      $\boldsymbol{B}_d^{(t+1)} = \mathrm{argmax}_{\boldsymbol{B}_d} \, \ell(\boldsymbol{\gamma}^{(t)}, \boldsymbol{B}_1^{(t+1)}, \ldots, \boldsymbol{B}_{d-1}^{(t+1)}, \boldsymbol{B}_d, \boldsymbol{B}_{d+1}^{(t)}, \ldots, \boldsymbol{B}_D^{(t)}, \boldsymbol{G}^{(t)})$
    end for
    $\boldsymbol{G}^{(t+1)} = \mathrm{argmax}_{\boldsymbol{G}} \, \ell(\boldsymbol{\gamma}^{(t)}, \boldsymbol{B}_1^{(t+1)}, \ldots, \boldsymbol{B}_D^{(t+1)}, \boldsymbol{G})$
    $\boldsymbol{\gamma}^{(t+1)} = \mathrm{argmax}_{\boldsymbol{\gamma}} \, \ell(\boldsymbol{\gamma}, \boldsymbol{B}_1^{(t+1)}, \ldots, \boldsymbol{B}_D^{(t+1)}, \boldsymbol{G}^{(t+1)})$
until $\ell(\boldsymbol{\theta}^{(t+1)}) - \ell(\boldsymbol{\theta}^{(t)}) < \epsilon$

Proposition 1. *Assume (i) the log-likelihood function ℓ is continuous, coercive, i.e., the set $\{\boldsymbol{\theta} : \ell(\boldsymbol{\theta}) \geq \ell(\boldsymbol{\theta}^{(0)})\}$ is compact, and bounded above, (ii) the objective function in each block update of Algorithm 1 is strictly concave, and (iii) the set of stationary points (modulo nonsingular transformation indeterminacy) of $\ell(\boldsymbol{\gamma}, \boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)$ are isolated. We have the following results.*

1. *(Global Convergence) The sequence $\boldsymbol{\theta}^{(t)} = (\boldsymbol{\gamma}^{(t)}, \boldsymbol{G}^{(t)}, \boldsymbol{B}_1^{(t)}, \ldots, \boldsymbol{B}_D^{(t)})$ generated by Algorithm 1 converges to a stationary point of $\ell(\boldsymbol{\gamma}, \boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)$.*

2. *(Local Convergence) Let $\boldsymbol{\theta}^{(\infty)} = (\boldsymbol{\gamma}^{(\infty)}, \boldsymbol{G}^{(\infty)}, \boldsymbol{B}_1^{(\infty)}, \ldots, \boldsymbol{B}_D^{(\infty)})$ be a strict local maximum of ℓ. The iterates generated by Algorithm 1 are locally attracted to $\boldsymbol{\theta}^{(\infty)}$ for $\boldsymbol{\theta}^{(0)}$ sufficiently close to $\boldsymbol{\theta}^{(\infty)}$.*

4   Statistical Theory

In this section we study the usual large n asymptotics of the proposed Tucker tensor regression. Regularization is treated in the next section for the small or moderate n cases. For simplicity, we drop the classical covariate $\boldsymbol{Z}$ in this section, but all the results can be straightforwardly extended to include $\boldsymbol{Z}$. We also remark that, although the usually limited sample size of neuroimging studies makes the large n asymptotics seem irrelevant, we still believe such an asymptotic investigation important, for several reasons. First, when the sample size n is considerably larger than the effective number of parameters p_{T}, the asymptotic study tells us that the model is consistently estimating the best Tucker structure approximation to the full array model in the sense of Kullback-Liebler distance. Second, the explicit formula

for score and information are not only useful for asymptotic theory but also for computation, while the identifiability issue has to be properly dealt with for the given model. Finally, the regular asymptotics can be of practical relevance, for instance, can be useful in a likelihood ratio type test in a replication study.

4.1 Score and Information

We first derive the score and information for the tensor regression model, which are essential for statistical estimation and inference. The following standard calculus notations are used. For a scalar function f, ∇f is the (column) gradient vector, $df = [\nabla f]^\mathsf{T}$ is the differential, and $d^2 f$ is the Hessian matrix. For a multivariate function $g : \mathbb{R}^p \mapsto \mathbb{R}^q$, $Dg \in \mathbb{R}^{p \times q}$ denotes the Jacobian matrix holding partial derivatives $\frac{\partial g_j}{\partial x_i}$. We start from the Jacobian and Hessian of the systematic part $\eta \equiv g(\mu)$ in (4).

Lemma 2.   *1. The gradient $\nabla\eta(\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \in \mathbb{R}^{\prod_d R_d + \sum_{d=1}^D p_d R_d}$ is*

$$\nabla\eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) = [\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \ \boldsymbol{J}_1 \ \boldsymbol{J}_2 \ \cdots \ \boldsymbol{J}_D]^\mathsf{T}(\mathrm{vec}\boldsymbol{X}),$$

where $\boldsymbol{J}_d \in \mathbb{R}^{\prod_{d=1}^D p_d \times p_d R_d}$ is the Jacobian

$$\boldsymbol{J}_d = D\boldsymbol{B}(\boldsymbol{B}_d) = \boldsymbol{\Pi}_d\{[(\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1)\boldsymbol{G}_{(d)}^\mathsf{T}] \otimes \boldsymbol{I}_{p_d}\} \tag{5}$$

and $\boldsymbol{\Pi}_d$ is the $(\prod_{d=1}^D p_d)$-by-$(\prod_{d=1}^D p_d)$ permutation matrix that reorders $\mathrm{vec}\boldsymbol{B}_{(d)}$ to obtain $\mathrm{vec}\boldsymbol{B}$, i.e., $\mathrm{vec}\boldsymbol{B} = \boldsymbol{\Pi}_d \mathrm{vec}\boldsymbol{B}_{(d)}$.

2. Let the Hessian $d^2\eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \in \mathbb{R}^{(\prod_d R_d + \sum_d p_d R_d) \times (\prod_d R_d + \sum_d p_d R_d)}$ be partitioned into four blocks $\boldsymbol{H}_{\boldsymbol{G},\boldsymbol{G}} \in \mathbb{R}^{\prod_d R_d \times \prod_d R_d}$, $\boldsymbol{H}_{\boldsymbol{G},\boldsymbol{B}} = \boldsymbol{H}_{\boldsymbol{B},\boldsymbol{G}}^\mathsf{T} \in \mathbb{R}^{\prod_d R_d \times \sum_d p_d R_d}$ and $\boldsymbol{H}_{\boldsymbol{B},\boldsymbol{B}} \in \mathbb{R}^{\sum_d p_d R_d \times \sum_d p_d R_d}$. Then $\boldsymbol{H}_{\boldsymbol{G},\boldsymbol{G}} = \boldsymbol{0}$, $\boldsymbol{H}_{\boldsymbol{G},\boldsymbol{B}}$ has entries

$$h_{(r_1, \ldots, r_D), (i_d, s_d)} = 1_{\{r_d = s_d\}} \sum_{j_d = i_d} x_{j_1, \ldots, j_D} \prod_{d' \neq d} \beta_{j_{d'}}^{(r_{d'})},$$

and $\boldsymbol{H}_{\boldsymbol{B},\boldsymbol{B}}$ has entries

$$h_{(i_d, r_d), (i_{d'}, r_{d'})} = 1_{\{d \neq d'\}} \sum_{j_d = i_d, j_{d'} = i_{d'}} x_{j_1, \ldots, j_D} \sum_{s_d = r_d, s_{d'} = r_{d'}} g_{s_1, \ldots, s_D} \prod_{d'' \neq d, d'} \beta_{j_{d''}}^{(s_{d''})}.$$

13

Furthermore, $\boldsymbol{H_{B,B}}$ can be partitioned in D^2 sub-blocks as

$$\begin{pmatrix} \boldsymbol{0} & * & * & * \\ \boldsymbol{H}_{21} & \boldsymbol{0} & * & * \\ \vdots & \vdots & \ddots & * \\ \boldsymbol{H}_{D1} & \boldsymbol{H}_{D2} & \cdots & \boldsymbol{0} \end{pmatrix}.$$

The elements of sub-block $\boldsymbol{H}_{dd'} \in \mathbb{R}^{p_d R_d \times p_{d'} R_{d'}}$ can be retrieved from the matrix

$$\boldsymbol{X}_{(dd')}(\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_{d'+1} \otimes \boldsymbol{B}_{d'-1} \otimes \cdots \otimes \boldsymbol{B}_1)\boldsymbol{G}_{(dd')}^{\mathsf{T}}.$$

$\boldsymbol{H_{G,B}}$ can be partitioned into D sub-blocks as $(\boldsymbol{H}_1, \ldots, \boldsymbol{H}_D)$. The sub-block $\boldsymbol{H}_d \in \mathbb{R}^{\prod_d R_d \times p_d R_d}$ has at most $p_d \prod_d R_d$ nonzero entries which can be retrieved from the matrix

$$\boldsymbol{X}_{(d)}(\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_{d+1} \otimes \boldsymbol{B}_{d-1} \otimes \cdots \otimes \boldsymbol{B}_1).$$

Let $\ell(\boldsymbol{B}_1, \ldots, \boldsymbol{B}_D | y, \boldsymbol{x}) = \ln p(y | \boldsymbol{x}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)$ be the log-density of GLM. Next result derives the score function, Hessian, and Fisher information of the Tucker tensor regression model.

Proposition 2. *Consider the tensor regression model defined by (4) and (4).*

1. *The score function (or score vector) is*

$$\nabla\ell(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) = \frac{(y - \mu)\mu'(\eta)}{\sigma^2}\nabla\eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \tag{6}$$

with $\nabla\eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)$ given in Lemma 2.

2. *The Hessian of the log-density ℓ is*

$$\begin{aligned} &H(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \\ =\ & -\left[\frac{[\mu'(\eta)]^2}{\sigma^2} - \frac{(y-\mu)\theta''(\eta)}{\sigma^2}\right]\nabla\eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)d\eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \\ &+ \frac{(y-\mu)\theta'(\eta)}{\sigma^2}d^2\eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D), \end{aligned} \tag{7}$$

with $d^2\eta$ defined in Lemma 2.

3. The Fisher information matrix is

$$
\begin{aligned}
& \boldsymbol{I}(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) \\
& = E[-H(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)] \\
& = \text{Var}[\nabla\ell(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)d\ell(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)] \\
& = \frac{[\mu'(\eta)]^2}{\sigma^2}[\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \ \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]^\mathsf{T}(\text{vec}\boldsymbol{X})(\text{vec}\boldsymbol{X})^\mathsf{T}[\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \ \boldsymbol{J}_1 \ldots \boldsymbol{J}_D] (8)
\end{aligned}
$$

Remark 2.1: For canonical link, $\theta = \eta$, $\theta'(\eta) = 1$, $\theta''(\eta) = 0$, and the second term of Hessian vanishes. For the classical GLM with linear systematic part ($D = 1$), $d^2\eta(\boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D)$ is zero and thus the third term of Hessian vanishes. For the classical GLM ($D = 1$) with canonical link, both second and third terms of the Hessian vanish and thus the Hessian is non-stochastic, coinciding with the information matrix.

4.2   Identifiability

The Tucker decomposition (3) is unidentifiable due to the nonsingular transformation indeterminacy. That is

$$
[\![\boldsymbol{G}; \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D]\!] = [\![\boldsymbol{G} \times_1 \boldsymbol{O}_1^{-1} \times \cdots \times_D \boldsymbol{O}_D^{-1}; \boldsymbol{B}_1\boldsymbol{O}_1, \ldots, \boldsymbol{B}_D\boldsymbol{O}_D]\!]
$$

for any nonsingular matrices $\boldsymbol{O}_d \in \mathbb{R}^{R_d \times R_d}$. This implies that the number of free parameters for a Tucker model is $\sum_d p_d R_d + \prod_d R_d - \sum_d R_d^2$, with the last term adjusting for nonsingular indeterminacy. Therefore the Tucker model is identifiable only in terms of the equivalency classes.

For asymptotic consistency and normality, it is necessary to adopt a specific constrained parameterization. It is common to impose the orthonormality constraint on the factor matrices $\boldsymbol{B}_d^\mathsf{T}\boldsymbol{B}_d = \boldsymbol{I}_{R_d}$, $d = 1, \ldots, D$. However the resulting parameter space is a manifold and much harder to deal with. We adopt an alternative parameterization that fixes the entries of the first R_d rows of $\boldsymbol{B}_d$ to be ones

$$
\mathcal{B} = \{[\![\boldsymbol{G}; \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D]\!] : \beta_{i_d}^{(r)} = 1, i_d = 1, \ldots, R_d, d = 1, \ldots, D\}.
$$

The formulae for score, Hessian and information in Proposition 2 require changes accordingly. The entries in the first R_d rows of $\boldsymbol{B}_d$ are fixed at ones and their corresponding entries,

rows and columns in score, Hessian and information need to be deleted. Choice of the restricted space $\mathcal{B}$ is obviously arbitrary, and excludes arrays with any entries in the first rows of $\boldsymbol{B}_d$ equal to zeros. However the set of such exceptional arrays has Lebesgue measure zero. In specific applications, subject knowledge may suggest alternative restrictions on the parameters.

Given a finite sample size, conditions for global identifiability of parameters are in general hard to obtain except in the linear case ($D = 1$). Local identifiability essentially requires linear independence between the "collapsed" vectors $[\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \ \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]^\mathsf{T} \text{vec} \boldsymbol{x}_i \in \mathbb{R}^{\sum_d p_d R_d + \prod_d R_d - \sum_d R_d^2}$.

Proposition 3 (Identifiability). *Given iid data points $\{(y_i, \boldsymbol{x}_i), i = 1, \ldots, n\}$ from the Tucker tensor regression model. Let $\boldsymbol{B}_0 \in \mathcal{B}$ be a parameter point and assume there exists an open neighborhood of $\boldsymbol{B}_0$ in which the information matrix has a constant rank. Then $\boldsymbol{B}_0$ is locally identifiable if and only if*

$$I(\boldsymbol{B}_0) = [\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \ \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]^\mathsf{T} \left[\sum_{i=1}^{n} \frac{\mu'(\eta_i)^2}{\sigma_i^2} (\text{vec} \, \boldsymbol{x}_i)(\text{vec} \, \boldsymbol{x}_i)^\mathsf{T} \right] [\boldsymbol{B}_D \otimes \cdots \otimes \boldsymbol{B}_1 \ \boldsymbol{J}_1 \ldots \boldsymbol{J}_D]$$

is nonsingular.

4.3 Asymptotics

The asymptotics for tensor regression follow from those for MLE or M-estimation. The key observation is that the nonlinear part of tensor model (4) is a degree-D polynomial of parameters and the collection of polynomials $\{\langle \boldsymbol{B}, \boldsymbol{X} \rangle, \boldsymbol{B} \in \mathcal{B}\}$ form a Vapnik-Červonenkis (VC) class. Then the classical uniform convergence theory applies (van der Vaart, 1998). For asymptotic normality, we need to establish that the log-likelihood function of tensor regression model is quadratic mean differentiable (Lehmann and Romano, 2005). A sketch of the proof is given in the Appendix.

Theorem 1. *Assume $\boldsymbol{B}_0 \in \mathcal{B}$ is (globally) identifiable up to permutation and the array covariates $\boldsymbol{X}_i$ are iid from a bounded underlying distribution.*

1. *(Consistency) The MLE is consistent, i.e., $\hat{\boldsymbol{B}}_n$ converges to $\boldsymbol{B}_0$ in probability, in following models. (1) Normal tensor regression with a compact parameter space $\mathcal{B}_0 \subset \mathcal{B}$.*

(2) Binary tensor regression. (3) Poisson tensor regression with a compact parameter space $\mathcal{B}_0 \subset \mathcal{B}$.

2. *(Asymptotic Normality) For an interior point $\boldsymbol{B}_0 \in \mathcal{B}$ with nonsingular information matrix $\boldsymbol{I}(\boldsymbol{B}_0)$ (8) and $\hat{\boldsymbol{B}}_n$ is consistent, $\sqrt{n}(\mathrm{vec}\hat{\boldsymbol{B}}_n - \mathrm{vec}\boldsymbol{B}_0)$ converges in distribution to a normal with mean zero and covariance matrix $\boldsymbol{I}^{-1}(\boldsymbol{B}_0)$.*

In practice it is rare that the true regression coefficient $\boldsymbol{B}_{\mathrm{true}} \in \mathbb{R}^{p_1 \times \cdots \times p_D}$ is exactly a low rank tensor. However the MLE of the rank-R tensor model converges to the maximizer of function $M(\boldsymbol{B}) = \mathbb{P}_{\boldsymbol{B}_{\mathrm{true}}} \ln p_{\boldsymbol{B}}$ or equivalently $\mathbb{P}_{\boldsymbol{B}_{\mathrm{true}}} \ln(p_{\boldsymbol{B}}/p_{\boldsymbol{B}_{\mathrm{true}}})$. In other words, the MLE consistently estimates the best approximation (among models in $\mathcal{B}$) of $\boldsymbol{B}_{\mathrm{true}}$ in the sense of Kullback-Leibler distance.

5   Regularized Estimation

Regularization plays a crucial role in neuroimaging analysis for several reasons. First, even after substantial dimension reduction by imposing a Tucker structure, the number of parameters p_{T} can still exceed the number of observations n. Second, even when $n > p_{\mathrm{T}}$, regularization could potentially be useful for stabilizing the estimates and improving the risk property. Finally, regularization is an effective way to incorporate prior scientific knowledge about brain structures. For instance, it may sometimes be reasonable to impose symmetry on the parameters along the coronal plane for MRI images.

In our context of Tucker regularized regression, there are two possible types of regularizations, one on the core tensor $\boldsymbol{G}$ *only*, and the other on both $\boldsymbol{G}$ and $\boldsymbol{B}_d$ *simultaneously*. Which regularization to use depends on the practical purpose of a scientific study. In this section, we illustrate the regularization on the core tensor, which simultaneously achieves sparsity in the number of outer products in Tucker decomposition (3) and shrinkage. Toward that purpose, we propose to maximize the regularized log-likelihood

$$\ell(\boldsymbol{\gamma}, \boldsymbol{G}, \boldsymbol{B}_1, \ldots, \boldsymbol{B}_D) - \sum_{r_1, \ldots, r_D} P_\eta(|g_{r_1, \ldots, r_D}|, \lambda),$$

where $P_\eta(|x|, \lambda)$ is a scalar penalty function, λ is the penalty tuning parameter, and η is an index for the penalty family. Note that the penalty term above only involves elements of the core tensor, and thus regularization on $\boldsymbol{G}$ only. This formulation includes a large class of

penalty functions, including power family (Frank and Friedman, 1993), where $P_\eta(|x|, \lambda) = \lambda|x|^\eta$, $\eta \in (0, 2]$, and in particular lasso (Tibshirani, 1996) ($\eta = 1$) and ridge ($\eta = 2$); elastic net (Zou and Hastie, 2005), where $P_\eta(|x|, \lambda) = \lambda[(\eta - 1)x^2/2 + (2 - \eta)|x|]$, $\eta \in [1, 2]$; SCAD (Fan and Li, 2001), where $\partial/\partial|x|P_\eta(|x|, \lambda) = \lambda \left\{ 1_{\{|x| \leq \lambda\}} + (\eta\lambda - |x|)_+/(\eta - 1)\lambda 1_{\{|x| > \lambda\}} \right\}$, $\eta > 2$; and MC+ penalty (Zhang, 2010), where $P_\eta(|x|, \lambda) = \left\{ \lambda|x| - x^2/(2\eta) \right\} 1_{\{|x| < \eta\lambda\}} + 0.5\lambda^2\eta 1_{\{|x| \geq \eta\lambda\}}$, among many others.

Two aspects of the proposed regularized Tucker regression, parameter estimation and tuning, deserve some discussion. For regularized estimation, it incurs only slight changes in Algorithm 1. That is, when updating $\boldsymbol{G}$, we simply fit a penalized GLM regression problem,

$$\boldsymbol{G}^{(t+1)} = \text{argmax}_{\boldsymbol{G}} \, \ell(\boldsymbol{\gamma}^{(t)}, \boldsymbol{B}_1^{(t+1)}, \ldots, \boldsymbol{B}_D^{(t+1)}, \boldsymbol{G}) - \sum_{r_1, \ldots, r_D} P_\eta(|g_{r_1, \ldots, r_D}|, \lambda),$$

for which many software packages exist. Our implementation utilizes an efficient MATLAB toolbox for sparse regression (Zhou et al., 2011). Other steps of Algorithm 1 remain unchanged. For the regularization to remain legitimate, we constrain the column norms of $\boldsymbol{B}_d$ to be one when updating factor matrices $\boldsymbol{B}_d$. For parameter tuning, one can either use the general cross validation approach, or employ Bayesian information criterion to tune the penalty parameter λ.

6 Numerical Study

We have carried out intensive numerical experiments to study the finite sample performance of the Tucker regression. Our simulations focus on three aspects: first, we demonstrate the capacity of the Tucker regression in identifying various shapes of signals; second, we study the consistency property of the method by gradually increasing the sample size; third, we compare the performance of the Tucker regression with the CP regression of Zhou et al. (2013). We also examine a real MRI imaging data to illustrate the Tucker downsizing and to further compare the two tensor models.

6.1 Identification of Various Shapes of Signals

In our first example, we demonstrate that the proposed Tucker regression model, though with substantial reduction in dimension, can manage to identify a range of two dimensional

signal shapes with varying ranks. In Figure 2, we list the 2D signals $\boldsymbol{B} \in \mathbb{R}^{64 \times 64}$ in the first row, along with the estimates by Tucker tensor models in the second to fourth rows with orders $(1, 1), (2, 2)$ and $(3, 3)$, respectively. Note that, since the orders along both dimensions are made equal, the Tucker model is to perform essentially the same as a CP model in this example, and the results are presented here for completeness. We will examine differences of the two models in later examples. The regular covariate vector $\boldsymbol{Z} \in \mathbb{R}^{5}$ and image covariate $\boldsymbol{X} \in \mathbb{R}^{64 \times 64}$ are randomly generated with all elements being independent standard normals. The response Y is generated from a normal model with mean $\mu = \boldsymbol{\gamma}^{\mathsf{T}}\boldsymbol{Z} + \langle \boldsymbol{B}, \boldsymbol{X} \rangle$ and variance $\mathrm{var}(\mu)/10$. The vector coefficient $\boldsymbol{\gamma} = \mathbf{1}_{5}$, and the coefficient array $\boldsymbol{B}$ is binary, with the signal region equal to one and the rest zero. Note that this problem differs from the usual edge detection or object recognition in imaging processing (Qiu, 2005, 2007). In our setup, all elements of the image $\boldsymbol{X}$ follow the same distribution. The signal region is defined through the coefficient matrix $\boldsymbol{B}$ and needs to be inferred from the relation between Y and $\boldsymbol{X}$ after adjusting for $\boldsymbol{Z}$. It is clearly see in Figure 2 that, the Tucker model yields a sound recovery of the true signals, even for those of high rank or natural shape, e.g., "disk" and "butterfly". We also illustrate in the plot the BIC criterion in Section 2.4.

6.2 Performance with Increasing Sample Size

In our second example, we continue to employ a similar model as in Figure 2 but with a three dimensional image covariate. The dimension of $\boldsymbol{X}$ is set as $p_1 \times p_2 \times p_3$, with $p_1 = p_2 = p_3 = 16$ and 32, respectively. The signal array $\boldsymbol{B}$ is generated from a Tucker structure, with the elements of core tensor $\boldsymbol{G}$ and the factor matrices $\boldsymbol{B}$'s all coming from independent standard normals. The dimension of the core tensor $\boldsymbol{G}$ is set as $R_1 \times R_2 \times R_3$, with $R_1 = R_2 = R_3 = 2, 5$, and 8, respectively. We gradually increase the sample size, starting with an n that is in hundred and no smaller than the degrees of freedom of the generating model. We aim to achieve two purposes with this example: first, we verify the consistency property of the proposed estimator, and second, we gain some practical knowledge about the estimation accuracy with different values of the sample size. Figure 3 summarizes the results. It is clearly seen that the estimation improves with the increasing sample size. Meanwhile, we observe that, unless the core tensor dimension is small, one would require a relatively large sample size to achieve a good estimation accuracy. This is not surprising though, considering
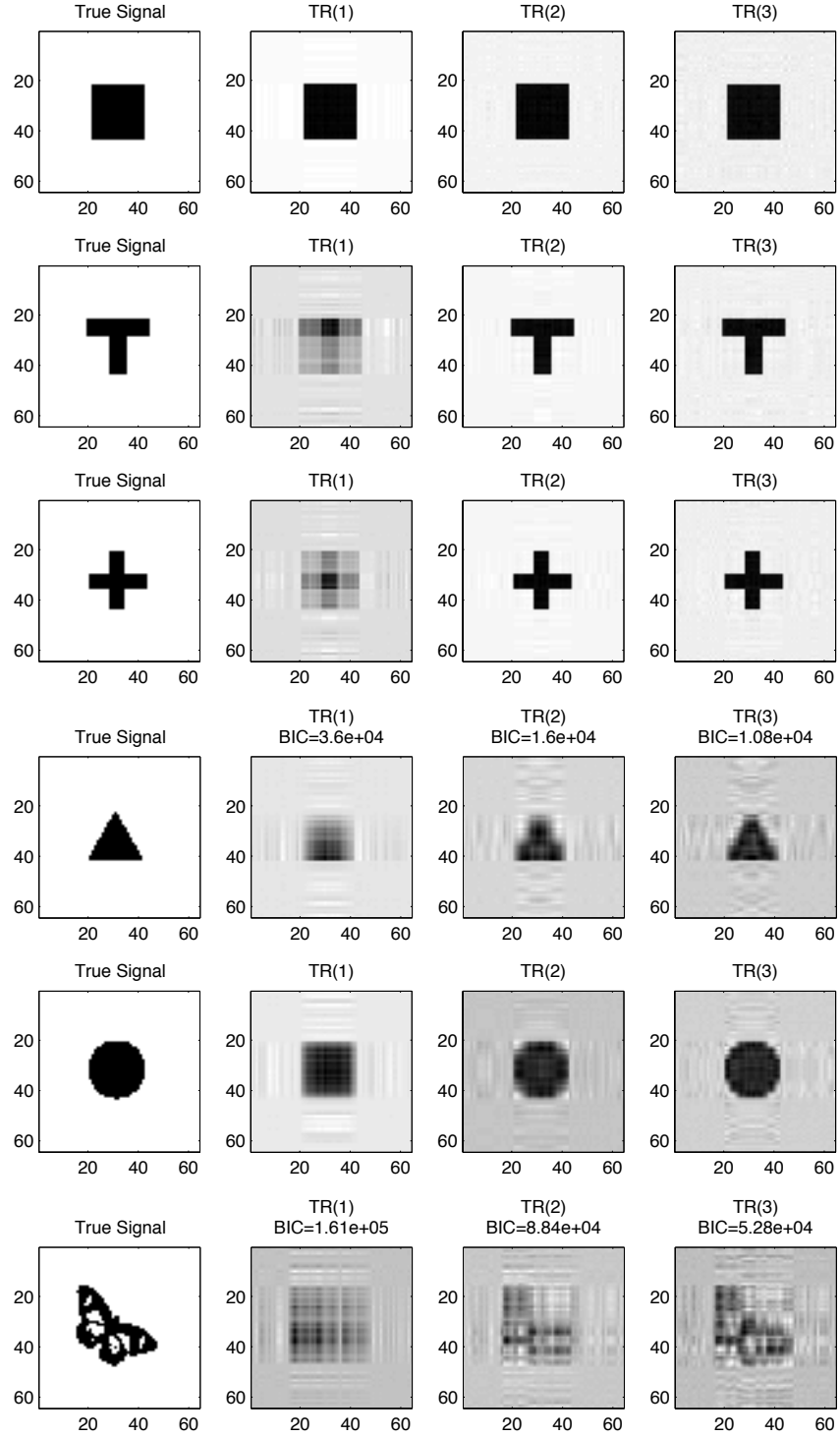
Figure 2: True and recovered image signals by Tucker regression. The matrix variate has size 64 by 64 with entries generated as independent standard normals. The regression coefficient for each entry is either 0 (white) or 1 (black). The sample size is 1000. TR(r) means estimate from the Tucker regression with an r-by-r core tensor.

20

the number of parameters of the model and that regularization is not employed here. The proposed tensor regression approach has been primarily designed for imaging studies with a reasonably large number of subjects. Recently, a number of such large-scale brain imaging studies are emerging. For instance, the Attention Deficit Hyperactivity Disorder Sample Initiative (ADHD, 2013) consists of over 900 participants from eight imaging centers with both MRI and fMRI images, as well as their clinical information. Another example is the Alzheimer's Disease Neuroimaging Initiative (ADNI, 2013) database, which accumulates over 3,000 participants with MRI, fMRI and genomics data. In addition, regularization discussed in Section 5 and the Tucker downsizing in Section 2.3 can both help improve estimation given a limited sample size.

6.3 Comparison of the Tucker and CP Models

In our third example, we focus on comparison between the Tucker tensor model with the CP tensor model of Zhou et al. (2013). We generate a normal response, and the 3D signal array $\boldsymbol{B}$ with dimensions p_1, p_2, p_3 and the d-ranks r_1, r_2, r_3. Here, the d-rank is defined as the column rank of the mode-d matricization $\boldsymbol{B}_{(d)}$ of $\boldsymbol{B}$. We set $p_1 = p_2 = p_3 = 16$ and 32, and $(r_1, r_2, r_3) = (5, 3, 3), (8, 4, 4)$ and $(10, 5, 5)$, respectively. The sample size is 2000. We fit a Tucker model with $R_d = r_d$, and a CP model with $R = \max r_d$, $d = 1, 2, 3$. We report in Table 2 the degrees of freedom of the two models under different setup, as well as the root mean squared error (RMSE) out of 100 data replications. It is seen that the Tucker model requires a smaller number of free parameters, while it achieves a more accurate estimation compared to the CP model. Such advantages come from the flexibility of the Tucker decomposition that permits different orders R_d along directions.

6.4 Attention Deficit Hyperactivity Disorder Data Analysis

We analyze the attention deficit hyperactivity disorder (ADHD) data from the ADHD-200 Sample Initiative (ADHD, 2013) to illustrate our proposed method as well as the Tucker downsizing. ADHD is a common childhood disorder and can continue through adolescence and adulthood. Symptoms include difficulty in staying focused and paying attention, difficulty in controlling behavior, and over-activity. The data set that we analyzed is part of the ADHD-200 Global Competition data sets. It was pre-partitioned into a training
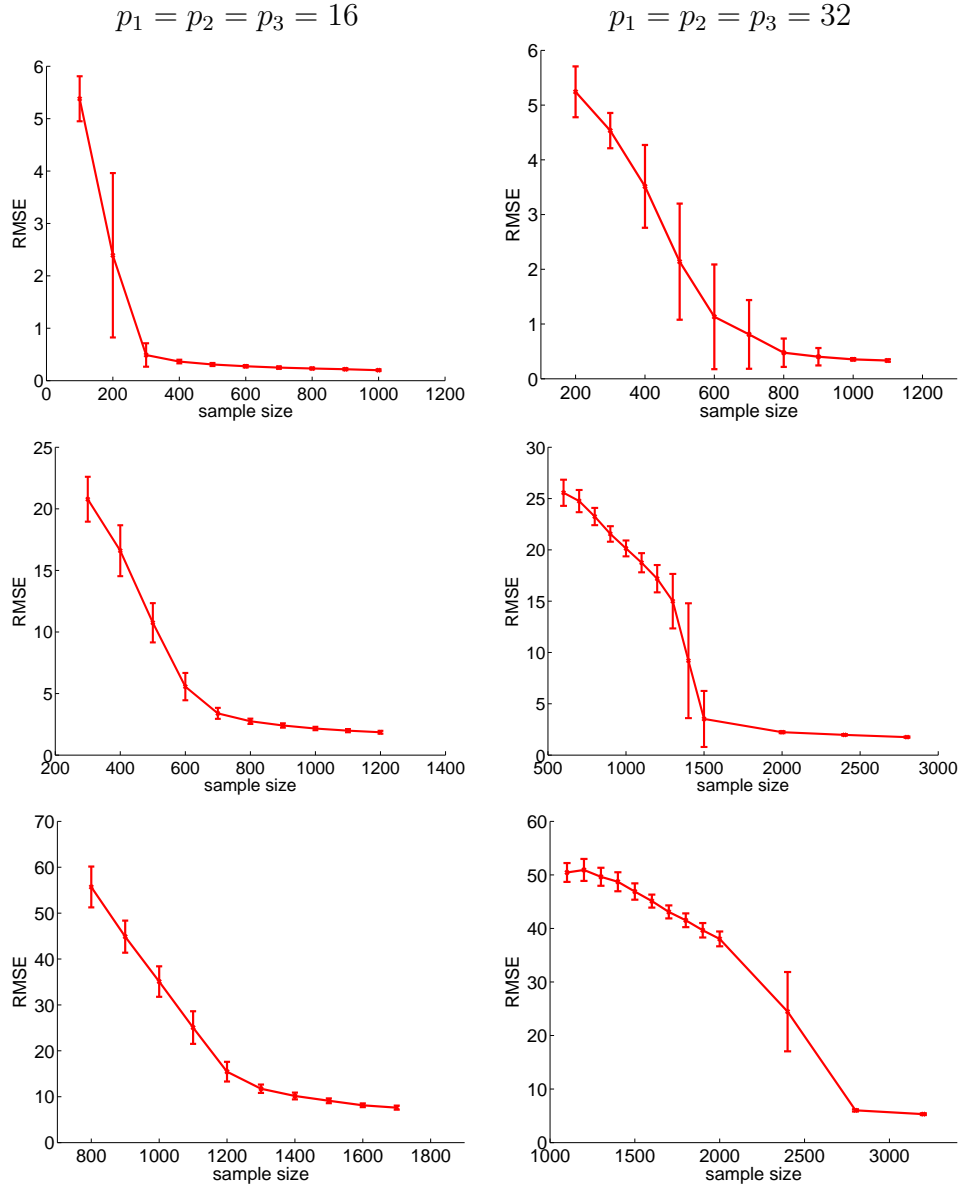
Figure 3: Root mean squared error (RMSE) of the tensor parameter estimate versus the sample size. Reported are the average and standard deviation of RMSE based on 100 data replications. Top: $R_1 = R_2 = R_3 = 2$; Middle: $R_1 = R_2 = R_3 = 5$; Bottom: $R_1 = R_2 = R_3 = 8$.

Table 2: Comparison of the Tucker and CP models. Reported are the average and standard deviation (in the parenthesis) of the root mean squared error, all based on 100 data replications.

Dimension	Criterion	Model	$(5, 3, 3)$	$(8, 4, 4)$	$(10, 5, 5)$
$16 \times 16 \times 16$	Df	Tucker	178	288	420
		CP	230	368	460
	RMSE	Tucker	0.202 (0.013)	0.379 (0.017)	0.728 (0.030)
		CP	0.287 (0.033)	1.030 (0.081)	2.858 (0.133)
$32 \times 32 \times 32$	Df	Tucker	354	544	740
		CP	470	752	940
	RMSE	Tucker	0.288 (0.013)	0.570 (0.023)	1.236 (0.045)
		CP	0.392 (0.046)	1.927 (0.172)	16.238 (3.867)

data of 770 subjects and a testing data of 197 subjects. We removed those subjects with missing observations or poor image quality, resulting in 762 training subjects and 169 testing subjects. In the training set, there were 280 combined ADHD subjects, 482 normal controls, and the case-control ratio is about 3:5. In the testing set, there were 76 combined ADHD subjects, 93 normal controls, and the case-control ratio is about 4:5. T1-weighted images were acquired for each subject, and were preprocessed by standard steps. The data we used is obtained from the Neuro Bureau after preprocessing (the Burner data, http://neurobureau.projects.nitrc.org/ADHD200/Data.html). In addition to the MRI image predictor, we also include the subjects' age and handiness as regular covariates. The response is the binary diagnosis status.

The original image size was $p_1 \times p_2 \times p_3 = 121 \times 145 \times 121$. We employ the Tucker downsizing in Section 2.3. More specifically, we first choose a wavelet basis for $\boldsymbol{B}_d \in \mathbb{R}^{p_d \times \tilde{p}_d}$, then transform the image predictor from $\boldsymbol{X}$ to $\tilde{\boldsymbol{X}} = [\![\boldsymbol{X}; \boldsymbol{B}_1^{\mathsf{T}}, \ldots, \boldsymbol{B}_D^{\mathsf{T}}]\!]$. We pre-specify the values of $\tilde{p}_d$'s that are about tenth of the original dimensions p_d, and equivalently, we fit a Tucker tensor regression with the image predictor dimension downsized to $\tilde{p}_1 \times \tilde{p}_2 \times \tilde{p}_3$. In our example, we have experimented with a set of values of $\tilde{p}_d$'s, and the results are qualitatively similar. We report two sets, $\tilde{p}_1 = 12, \tilde{p}_2 = 14, \tilde{p}_3 = 12$, and $\tilde{p}_1 = 10, \tilde{p}_2 = 12, \tilde{p}_3 = 10$. We have also experimented with the Haar wavelet basis (Daubechies D2) and the Daubechies D4 wavelet basis, which again show similar qualitative patterns.

For $\tilde{p}_1 = 12, \tilde{p}_2 = 14, \tilde{p}_3 = 12$, we fit a Tucker tensor model with $R_1 = R_2 = R_3 = 3$,

Table 3: ADHD testing data misclassification error.

Basis	Reduced dimension	Reg-Tucker	Reg-CP	Tucker	CP
Haar (D2)	$12 \times 14 \times 12$	0.361	0.367	0.379	0.438
	$10 \times 12 \times 10$	0.343	0.390	0.379	0.408
Daubechies (D4)	$12 \times 14 \times 12$	0.337	0.385	0.385	0.414
	$10 \times 12 \times 10$	0.320	0.396	0.367	0.373

resulting in 114 free parameters, and fit a CP tensor model with $R = 4$, resulting in 144 free parameters. For $\tilde{p}_1 = 10, \tilde{p}_2 = 12, \tilde{p}_3 = 10$, we fit a Tucker tensor model with $R_1 = R_2 = 2$ and $R_3 = 3$, resulting in 71 free parameters, and fit a CP tensor model with $R = 4$, resulting in 120 free parameters. We have chosen those orders based on the following considerations. First, the number of free parameters of the Tucker and CP models are comparable. Second, at each step of GLM model fit, we ensure that the ratio between the sample size n and the number of parameters under estimation in that step $\tilde{p}_d \times R_d$ satisfies a heuristic rule of greater than two in normal models and greater than five in logistic models. In the Tucker model, we also ensure the ratio between n and the number of parameters in the core tensor estimation $\prod_d R_d$ satisfies this rule. We note that this selection of Tucker orders is heuristic; however, it seems to be a useful guideline especially when the data is noisy. We also fit a regularized Tucker model and a regularized CP model with the same orders, while the penalty parameter is tuned based on 5-fold cross validation of the training data.

We evaluate each model by comparing the misclassification error rate on the independent testing set. The results are shown in Table 3. We see from the table that, the regularized Tucker model performs the best, which echoes the findings in our simulations above. We also remark that, considering the fact that the ratio of case-control is about 4:5 in the testing data, the misclassification rate from 0.32 to 0.36 achieved by the regularized Tucker model indicates a fairly sound classification accuracy. On the other hand, we note that, a key advantage of our proposed approach is its capability of suggesting a useful model rather than the classification accuracy per se. This is different from black-box type machine learning based imaging classifiers.

It is also of interest to compare the run times of the two tensor model fittings. We record the run times of fitting the Tucker and CP models with the ADHD training data in Table

Table 4: ADHD model fitting run time (in seconds).

Basis	Reduced dimension	Reg-Tucker	Reg-CP	Tucker	CP
Haar (D2)	$12 \times 14 \times 12$	3.68	4.39	31.25	22.43
	$10 \times 12 \times 10$	1.36	2.79	9.08	25.10
Daubechies (D4)	$12 \times 14 \times 12$	3.30	2.18	16.87	26.34
	$10 \times 12 \times 10$	1.92	1.90	9.96	17.10

4. They are comparable.

7   Discussion

We have proposed a tensor regression model based on the Tucker decomposition. Including the CP tensor regression (Zhou et al., 2013) as a special case, Tucker model provides a more flexible framework for regression with imaging covariates. We develop a fast estimation algorithm, a general regularization procedure, and the associated asymptotic properties. In addition, we provide a detailed comparison, both analytically and numerically, of the Tucker and CP tensor models.

In real imaging analysis, the signal hardly has an exact low rank. On the other hand, given the limited sample size, a low rank estimate often provides a reasonable approximation to the true signal. This is why the low rank models such as the Tucker and CP could offer a sound recovery of even a complex signal.

The tensor regression framework established in this article is general enough to encompass a large number of potential extensions, including but not limited to imaging multi-modality analysis, imaging classification, and longitudinal imaging analysis. These extensions consist of our future research.

References

ADHD (2013). The ADHD-200 sample. `http://fcon_1000.projects.nitrc.org/indi/` `adhd200/`. [Online; accessed 03-2013].

ADNI (2013). Alzheimer's disease neuroimaging initiative. `http://adni.loni.ucla.edu`. [Online; accessed 03-2013].

Allen, G., Grosenick, L., and Taylor, J. (2011). A generalized least squares matrix decomposition. *Rice University Technical Report No. TR2011-03*, arXiv:1102:3074.

Aston, J. A. and Kirch, C. (2012). Estimation of the distribution of change-points with application to fmri data. *Annals of Applied Statistics*, 6:1906–1948.

Blankertz, B., Curio, G., and Müller, K.-R. (2001). Classifying single trial EEG: Towards brain computer interfacing. In *NIPS*, pages 157–164.

Caffo, B., Crainiceanu, C., Verduzco, G., Joel, S., S.H., M., Bassett, S., and Pekar, J. (2010). Two-stage decompositions for the analysis of functional connectivity for fMRI with application to Alzheimer's disease risk. *Neuroimage*, 51(3):1140–1149.

Chen, S. S., Donoho, D. L., and Saunders, M. A. (2001). Atomic decomposition by basis pursuit. *SIAM Rev.*, 43(1):129–159.

Crainiceanu, C. M., Caffo, B. S., Luo, S., Zipunnikov, V. M., and Punjabi, N. M. (2011). Population value decomposition, a framework for the analysis of image populations. *J. Amer. Statist. Assoc.*, 106(495):775–790.

de Leeuw, J. (1994). Block-relaxation algorithms in statistics. In *Information Systems and Data Analysis*, pages 308–325. Springer, Berlin.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.*, 96(456):1348–1360.

Frank, I. E. and Friedman, J. H. (1993). A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135.

Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., and Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293(5539):2425–2430.

Hoff, P. (2011). Hierarchical multilinear models for multiway data. *Computational Statistics and Data Analysis*, 55:530–543.

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500.

Kontos, D., Megalooikonomou, V., Kontos, D., Faloutsos, C., Megalooikonomou, V., Ghubade, N., and Faloutsos, C. (2003). Detecting discriminative functional MRI activation patterns using space filling curves. In *in Proc. of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC*, pages 963–967. Springer-Verlag.

LaConte, S., Strother, S., Cherkassky, V., Anderson, J., and Hu, X. (2005). Support vector machines for temporal classification of block design fMRI data. *Neuroimage*, 26:317–329.

Lange, K. (2010). *Numerical Analysis for Statisticians.* Statistics and Computing. Springer, New York, second edition.

Lehmann, E. L. and Romano, J. P. (2005). *Testing Statistical Hypotheses.* Springer Texts in Statistics. Springer, New York, third edition.

McCullagh, P. and Nelder, J. A. (1983). *Generalized Linear Models.* Monographs on Statistics and Applied Probability. Chapman & Hall, London.

Mckeown, M. J., Makeig, S., Brown, G. G., Jung, T.-P., Kindermann, S. S., Kindermann, R. S., Bell, A. J., and Sejnowski, T. J. (1998). Analysis of fMRI data by blind separation into independent spatial components. *Human Brain Mapping*, 6:160–188.

Mitchell, T. M., Hutchinson, R., Niculescu, R. S., Pereira, F., Wang, X., Just, M., and Newman, S. (2004). Learning to decode cognitive states from brain images. *Machine Learning*, 57:145–175.

Qiu, P. (2005). *Image Processing and Jump Regression Analysis.* Wiley series in probability and statistics. John Wiley.

Qiu, P. (2007). Jump surface estimation, edge detection, and image restoration. *Journal of the American Statistical Association*, 102:745–756.

Ramsay, J. O. and Silverman, B. W. (2005). *Functional Data Analysis.* Springer-Verlag, New York.

Reiss, P. and Ogden, R. (2010). Functional generalized linear models with images as predictors. *Biometrics*, 66:61–69.

Shinkareva, S. V., Ombao, H. C., Sutton, B. P., Mohanty, A., and Miller, G. A. (2006). Classification of functional brain images with a spatio-temporal dissimilarity map. *NeuroImage*, 33(1):63–71.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288.

van der Vaart, A. W. (1998). *Asymptotic Statistics*, volume 3 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *Ann. Statist.*, 38(2):894–942.

Zhou, H., Armagan, A., and Dunson, D. (2011). Path following and empirical Bayes model selection for sparse regressions. *arXiv:1201.3528*.

Zhou, H., Li, L., and Zhu, H. (2013). Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, In press(arXiv:1203.3209).

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 67(2):301–320.